

LA-UR-17-20239

Approved for public release; distribution is unlimited.

Title: Large-Scale Inverse Model Analyses Employing Fast Randomized Data Reduction

Author(s): Lin, Youzuo
Le, Ellen
O'Malley, Daniel
Vesselinov, Velimir Valentinov
Bui, Tan

Intended for: Water Resources Research

Issued: 2017-01-13

Disclaimer:

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

₁ Large-Scale Inverse Model Analyses Employing Fast ₂ Randomized Data Reduction

Youzuo Lin¹, Ellen B. Le², Daniel O'Malley¹, Velimir V. Vesselinov¹, and

Tan Bui-Thanh²

Corresponding author: Youzuo Lin, Earth and Environment Science Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA. (ylin@lanl.gov)

¹Earth and Environmental Sciences
Division, Los Alamos National Laboratory,
Los Alamos, New Mexico, USA.

²Institute for Computational Sciences and
Engineering, The University of Texas at
Austin, Austin, Texas, USA.

Abstract.

When the number of observations is large, it is computationally challenging to apply hydraulic inverse modeling techniques. We have developed a new, computationally-efficient inverse modeling technique for solving inverse problems with a large number of observations (e.g. on the order of 10^6 or greater). Our method, which we call the randomized geostatistical approach (RGA), is built upon the principal component geostatistical approach (PCGA) developed by Kitanidis and others. We employ a data reduction technique in combination with the PCGA method to improve the computational efficiency and reduce the memory usage. Specifically, we employ a randomized numerical linear algebra technique to effectively reduce the dimension of the observations without losing the information content needed for the inverse analysis. Our algorithm is coded in Julia and implemented in the MADS open-source high-performance computational framework (<http://mads.lanl.gov>). We apply our new inverse modeling method to invert for a synthetic transmissivity field. By comparing with original PCGA method, our method yields a much more efficient computational cost when the number of observation is large. Most importantly, our method is capable of solving for inverse problems that are larger than it is possible with the standard PCGA approach. Therefore, our new inverse modeling method is a powerful tool for characterizing subsurface heterogeneity for large-scale real-world problems.

1. Introduction

The permeability of a porous medium is of great importance for predicting flow and transport of fluids and contaminants in the subsurface [Carrera and Neuman, 1986; Sun, 1994; Carrera et al., 2005]. A well-understood distribution of permeability heterogeneity can be crucial for many different subsurface applications such as (1) forecasting production performance of geothermal reservoirs, (2) extracting oil and gas, (3) estimating pathways of subsurface contaminant transport, and many others.

Various hydraulic inversion methods have been proposed and developed to obtain subsurface permeability [Neuman and Yakowitz, 1979; Neuman et al., 1980; Carrera and Neuman, 1986; Sun, 1994; Kitanidis, 1997a; Zhang and Yeh, 1997; Carrera et al., 2005], of which the geostatistical inversion is the most widely used [Kitanidis, 1995; Zhang and Yeh, 1997; Kitanidis, 1997a, b; Vesselinov et al., 2001a]. The geostatistical inversion can be more advantageous than many other subsurface inverse modeling methods in that it can not only provide uncertainty estimates, but also be suitable for data fusion [Vesselinov et al., 2001a, b; Illman et al., 2015; Yeh and Simunek, 2002]. However, as pointed out in Vesselinov et al. [2001b] and Illman et al. [2015], one drawback of the geostatistical inversion method is its high computational cost when the number of observations is large and the model is highly parameterized. In recent years, with the help of regularization techniques [Tarantola, 2005; Engl et al., 1996], there is a trend to increase the number of model parameters [Hunt et al., 2007]. It has been discussed in many references that these highly parameterized models have great potential for characterizing subsurface heterogeneity [Tonkin and Doherty, 2005; Hunt et al., 2007]. Meanwhile, as the theory

and computational tools related to characterization of geologic subsurface quickly moves into the new era of “big data”, many existing methodologies are facing the challenges of handling large number of unknown model parameters and large number of observations. Therefore, it becomes important to address the theoretical and computational issues of the geostatistical inversion methods.

The costs related to application of the geostatistical inversion methods comes from two folds: the computational cost and the memory cost. A number of computational techniques have been proposed and developed to alleviate the expensive costs of both the computation [Saibaba and Kitanidis, 2012; Liu et al., 2013; Ambikasaran et al., 2013; Constantine et al., 2014; Liu et al., 2014; Lee and Kitanidis, 2014; Lin et al., 2016] and memory [Saibaba and Kitanidis, 2012; Kitanidis and Lee, 2014; Lee and Kitanidis, 2014]. Among those references, some target for both computation and memory costs [Saibaba and Kitanidis, 2012; Kitanidis and Lee, 2014; Lee and Kitanidis, 2014].

In particular, one major direction to reduce the computational cost is based on the subspace approximation, i.e., solving a small-size approximated problem residing in a lower-dimensional subspace to save the computational cost. Several types of subspaces have been utilized in the references including principle components subspace [Kitanidis and Lee, 2014; Lee and Kitanidis, 2014; Tonkin and Doherty, 2005], Krylov subspace [Lin et al., 2016; Liu et al., 2014; Saibaba and Kitanidis, 2012], subspace spanned by reduced-order model [Liu et al., 2014], hierarchical matrix decomposition [Ambikasaran et al., 2013; Saibaba and Kitanidis, 2012], and active subspace [Constantine et al., 2014].

In geostatistical inversion methods, a majority of the memory is used in storing the matrices, such as Jacobian matrix and covariance matrix. In situations with a large

number of measurements and model parameters, it is prohibitively expensive to store these matrices. To overcome the memory issues, researchers have developed either some matrix-free or low-rank approximation methods. Specifically, in the work of *Kitanidis and Lee* [2014]; *Lee and Kitanidis* [2014] and *Saibaba and Kitanidis* [2012], a matrix-free Jacobian is developed to approximate the multiplication of Jacobian matrix with a vector by finite-difference operations. To further reduce the storage cost of the covariance matrices, various low-rank matrix approximation techniques have been developed, such as hierarchical decomposition [*Ambikasaran et al.*, 2013; *Saibaba and Kitanidis*, 2012] and principal component decomposition [*Kitanidis and Lee*, 2014; *Lee and Kitanidis*, 2014].

Randomized algorithms have received a great deal of attention in recent years [*Drineas and Mahoney*, 2016]. Randomized algorithms can be seen as either sampling or projection procedures [*Mahoney*, 2011]. Its main idea is to construct a sketching matrix of an input matrix. The matrix is usually a smaller matrix, which yields a good approximation and represents the essential information of the original input. Therefore, the sketching matrix can be applied as a surrogate for the original to compute quantities of interest [*Drineas and Mahoney*, 2016]. Randomized algorithms have been successfully applied to various scientific and engineering domains, such as scientific computation and numerical linear algebra [*Le et al.*, 2015; *Meng and Mahoney*, 2014; *Drineas et al.*, 2011; *Rokhlin and Tygert*, 2008], seismic full-waveform inversion and tomography [*Moghaddam et al.*, 2013; *Krebs et al.*, 2009], and medical imaging [*Huang et al.*, 2016; *Wang et al.*, 2015; *Zhang et al.*, 2012], etc.

Here, we present a new geostatistical inversion method employing a randomization-based data reduction technique to reduce both the computation and memory costs. Ran-

domization techniques allow the possibility to generate a surrogate system while reducing the data dimension. We employ Gaussian projection to produce the sketching matrix [Johnson and Lindenstrauss, 1984] and further apply it to the geostatistical inversion. With the new sketch system generated, we employ a direct linear solver to obtain the solution of the surrogate problem. Through our numerical cost analysis presented in this paper, we show that using our techniques, our new randomized geostatistical inversion method improves the computational efficiency and reduces memory cost significantly.

To evaluate the performance of our algorithm, we test our new randomized geostatistical inversion method to solve for a transmissivity field from observations of hydraulic head. The hydraulic heads were “observed” from the solution of the groundwater equation using a reference transmissivity field at a number of observation points (monitoring wells). We implement our algorithm in Julia [Bezanson *et al.*, 2014] as part of the MADS open-source high-performance computational framework [Vesselinov *et al.*, 2015]. By comparing with the results obtained using the conventional geostatistical inversion method, we show that our method significantly reduces the computational and memory costs while maintaining the accuracy of the inversion results.

In the following sections, we first briefly describe the fundamentals of inverse modeling and geostatistical inversion methods (Sec. 2). We then develop and discuss a randomized geostatistical inversion method (Sec. 3). We further elaborate on the computational and memory costs of our method (Sec. 4). We then apply our method to test problems and discuss the results (Sec. 5). Finally, concluding remarks are presented in Sec. 6.

2. Theory

2.1. Inverse Modeling

We consider a transient groundwater flow equation. The forward modeling problem can be written as

$$\mathbf{h} = f(T) + \varepsilon, \quad (1)$$

where \mathbf{h} is the hydraulic head, T is the transmissivity and $f(T)$ is the forward operator mapping from the transmissivity to the hydraulic head and ε is a term representing the additive noise and following the distribution of

$$\varepsilon \sim N(0, R), \quad (2)$$

where R is the error covariance matrix. The operator $f(\cdot)$ is nonlinear in that the map from the model parameters, T , to the state variable h is not a linear map.

Correspondingly, the problem of hydrogeologic inverse modeling is to estimate the transmissivity provided with available measurements. Usually, such a problem is posed as a minimization problem

$$\hat{\mathbf{m}} = \arg \min_{\mathbf{m}} \{ \|\mathbf{d} - f(\mathbf{m})\|_2^2 \}, \quad (3)$$

where \mathbf{d} represents a recorded hydraulic head dataset and \mathbf{m} is the model parameter, $\|\mathbf{d} - f(\mathbf{m})\|_2^2$ measures the data misfit, and $\|\cdot\|_2$ stands for the L_2 norm. Solving Eq. (3) yields a model $\hat{\mathbf{m}}$ that minimizes the mean-squared difference between observed and synthetic data. However inverse problems formulated via Eq. (3) are often severely ill-posed. Moreover, because of the nonlinearity of the forward modeling operator f , the solution of the inverse problem may be non-unique where multiple minima of the misfit function might provide acceptable inverse solutions. Regularization techniques can be used

to address the non-uniqueness of the solution and reduce the ill-posedness of the inverse problem.

A general regularization term incorporated with Eq. (3) can be posed as [Vogel, 2002; Hansen, 1998]

$$\hat{\mathbf{m}} = \arg \min_{\mathbf{m}} \{l(\mathbf{m})\} \quad (4)$$

$$= \arg \min_{\mathbf{m}} \{ \|\mathbf{d} - f(\mathbf{m})\|_2^2 + \lambda \mathcal{R}(\mathbf{m}) \}, \quad (5)$$

where $\mathcal{R}(\mathbf{m})$ is a general regularization term and the parameter λ is the regularization parameter, which controls the amount of regularization in the inversion.

2.2. Geostatistical Inverse Modeling

To further account for the errors in the observations and the model, we follow the work in Kitanidis and Lee [2014] and Lee and Kitanidis [2014], and employ the generalized least squares that weights the data misfit and regularization terms in Eq. (5) using covariance matrices

$$\begin{aligned} \hat{\mathbf{m}} &= \arg \min_{\mathbf{m}} \{g(\mathbf{m})\} \\ &= \arg \min_{\mathbf{m}} \{ \|\mathbf{d} - f(\mathbf{m})\|_R^2 + \lambda \mathcal{R}(\mathbf{m}) \}, \end{aligned} \quad (6)$$

The weighted data misfit and regularization terms are defined as

$$\|\mathbf{d} - f(\mathbf{m})\|_R^2 = (\mathbf{d} - f(\mathbf{m}))' R^{-1} (\mathbf{d} - f(\mathbf{m})), \quad (7)$$

and

$$\mathcal{R}(\mathbf{m}) = (\mathbf{m} - (X\beta))' Q^{-1} (\mathbf{m} - (X\beta)), \quad (8)$$

where X is a drift (trend) matrix, Q is the covariance matrix of the model parameters and R is defined in Eq. (2). The regularization term in Eq. (8) is Tikhonov regularization, which is commonly used [Vogel, 2002; Hansen, 1998].

With the Jacobian matrix, H , of the forward modeling operator f defined as

$$H = \left. \frac{\partial f}{\partial \mathbf{m}} \right|_{\mathbf{m}=\bar{\mathbf{m}}}, \quad (9)$$

we will have the linearized function of the forward modeling operator f as

$$f(\hat{\mathbf{m}}) \approx f(\bar{\mathbf{m}}) + H(\hat{\mathbf{m}} - \bar{\mathbf{m}}), \quad (10)$$

where $\hat{\mathbf{m}}$ is the current solution and $\bar{\mathbf{m}}$ is the previous solution.

According to Kitanidis [1997b] and Nowak and Cirpka [2004], the current solution $\hat{\mathbf{m}}$ in Eq. (10) is given as

$$\hat{\mathbf{m}} = X\boldsymbol{\beta} + QH^T\boldsymbol{\xi}, \quad (11)$$

where the vectors of $\boldsymbol{\beta}$ and $\boldsymbol{\xi}$ are solutions to the linear system below

$$\begin{bmatrix} HQH^T + R & HX \\ (HX)^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\xi} \\ \boldsymbol{\beta} \end{bmatrix} = \begin{bmatrix} \mathbf{y} - f(\bar{\mathbf{m}}) + H\bar{\mathbf{m}} \\ 0 \end{bmatrix}. \quad (12)$$

The Jacobian matrix of H in Eq. (12) is the most computational and memory demanding. Various techniques are employed to address this issue. With respect to the definition of the Jacobian matrix as given in Eq. (10), one approach to bypass the expensive explicit construction of the Jacobian matrix is to use its finite difference approximation [Lee and Kitanidis, 2014; Kitanidis and Lee, 2014; Liu et al., 2014], i.e.

$$H\mathbf{x} \approx \frac{1}{\delta} [f(\mathbf{x} + \delta\mathbf{x}) - f(\mathbf{x})], \quad (13)$$

where \mathbf{x} is a n -dimensional vector and δ is the finite difference interval. Another computational technique to reduce the expensive cost of Jacobian matrix construction is to employ

the hierarchical representation of the Jacobian matrix [Saibaba and Kitanidis, 2012]. The hierarchical representation of a matrix is to split the given matrix into a hierarchy of rectangular blocks and approximate each of the blocks by a low-rank matrix [Saibaba and Kitanidis, 2012; Bebendorf, 2008; Borm et al., 2003].

With the Jacobian matrix obtained approximately, two main categories of numerical methods have been developed to solve the above linear system in Eq. (12). One is based on direct solvers [Lee and Kitanidis, 2014; Kitanidis and Lee, 2014] and the other is based on the iterative solvers [Liu et al., 2014; Saibaba and Kitanidis, 2012; Nowak and Cirpka, 2004]. Direct solvers are mostly used in the situations when size of problems ranges from small to medium scale and the system matrix in Eq. (12) can be therefore explicitly constructed [Lee and Kitanidis, 2014; Kitanidis and Lee, 2014]. As pointed out in Lee and Kitanidis [2014], direct solvers can be used to solve dense linear systems of dimension up to $n \sim \mathcal{O}(10^4)$. On the other hand, for large-scale problems (dimension $n > \mathcal{O}(10^4)$), non-standard matrix representations must be used, and Krylov-subspace based iterative solvers such as GMRES [Saad and Schultz, 1986] or MINRES [Paige and Saunders, 1975] are favored over direct methods to solve Eq. (12) [Liu et al., 2014; Saibaba and Kitanidis, 2012].

Utilization of direct solvers or iterative solvers to solve the problem in Eq. (12) can be memory bound. Such a limitation can significantly reduce the computational efficiency when a large number of measurements are available. In particular, it can be observed from Eq. (12) that the number of the equations is on the same order as the number of the measurements. In many subsurface applications, it is increasingly common to calibrate models using a very large number of observations. As an example, Figure 1 illustrates

the cumulative number of water-level measurements as a function of time collected at the Los Alamos National Laboratory site. These data provide important information about hydrogeologic site conditions and are included in various model analyses. The data are characterized by periodic, rapid increases in the rate of data collection which has produced a large data set that currently contains $\mathcal{O}(10^7)$ observations. Employing the computational techniques mentioned above to solve linear systems of such a scale is beyond the computability and storage capacity of any methods regardless of the choice of direct or iterative solvers. As pointed out in *Kitanidis and Lee* [2014], the developed computational methodologies work best for problems with a modest number of observations. Therefore, there is a need to develop computational methods that would allow an efficient solution of Eq. (12) with a large number of measurements. In the next section, we will describe one approach to reduce the dimensionality of the data while maintaining the accuracy of the inverse results.

3. Randomized Geostatistical Inverse Modeling

3.1. Randomized Geostatistical Approach

We develop a new randomized geostatistical inversion method to reduce the data dimensionality and maintain the accuracy of the inversion result. The basic idea of this approach is to construct a sketching matrix, S , then replace the data, \mathbf{d} , with $S\mathbf{d}$, replace the forward model, $f(T)$, with $Sf(T)$, and the additive noise, ϵ , with $S\epsilon$; and use the PCGA method to perform the calibration. By multiplying all these vectors by S , we reduce the dimensionality (S has many columns, but not that many rows). At a high-level, multiplying by the sketching matrix solves the problems associated with a high-dimensional observation space and the PCGA method solves the problems associated

with a high-dimensional parameter space. By combining these methods, we solve both problems. Additionally, if a PCGA implementation is available, the randomized geostatistical approach is extremely easy to implement in high-level languages such as Julia, Matlab and Python (our Julia implementation consists of 3 lines of code).

The misfit function of the randomized geostatistical inversion is given by

$$\hat{\mathbf{m}} = \arg \min_{\mathbf{m}} \{ \|S\mathbf{d} - Sf(\mathbf{m})\|_2^2 + \lambda \mathcal{R}'(\mathbf{m}) \}, \quad (14)$$

where $S \in \mathcal{R}^{k_{\text{red}} \times n}$ is the sketching matrix and $k_{\text{red}} \ll n$ is the tunable reduced dimension.

The sketching matrix is also referred to as a Johnson-Lindenstrauss Transform [Kane and Nelson, 2014; Woodruff, 2014; Mahoney, 2011; Dasgupta et al., 2010; Clarkson and Woodruff, 2009; Sarlos, 2006]. With the new misfit function defined in Eq. (14) and

following a similar derivation as in the previous section, we will have a randomization

linear system below

$$\begin{bmatrix} SHQH^T S^T + R' & SHX \\ (SHX)^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\xi} \\ \boldsymbol{\beta} \end{bmatrix} = \begin{bmatrix} S(\mathbf{y} - f(\bar{\mathbf{m}}) + H\bar{\mathbf{m}}) \\ 0 \end{bmatrix}. \quad (15)$$

At this point, we need to specify R' . As discussed above, the forward modeling can be formulated as

$$S\mathbf{h} = Sf(T) + S\epsilon, \quad (16)$$

we can therefore derivative the data covariance matrix R' in Eq. (15) as

$$R' = \mathbb{E}[S\epsilon(S\epsilon)^T] = S\mathbb{E}[(\mathbf{h} - f(T))(\mathbf{h} - f(T))^T]S^T = SRS^T. \quad (17)$$

With the randomized linear system given in Eq. (15) and the covariance matrix in

Eq. (17), we will have the corresponding solution iterate, which shares a similar ex-

pression to the one given in Eq. (11)

$$\hat{\mathbf{m}} = X\boldsymbol{\beta} + QH^T S^T \boldsymbol{\xi}. \quad (18)$$

3.2. Selection of the Sketching Matrix

The sketching matrix S in Eq. (14) can be critical in reducing the data dimensionality and preserving the solution accuracy. The role of sketching matrix can be seen as preconditioning the input data to spread out or uniformize the information contained [Drineas and Mahoney, 2016]. With an appropriately selected sketching matrix, the solution to Eq. (14) yields high accuracy to the one of the original problem in Eq. (3).

Theoretically, preserving of the accuracy of the solution using much reduced data set is based upon the Johnson-Lindenstrauss Lemma, which was first proved in the 1980's [Johnson and Lindenstrauss, 1984]. It was pointed out by Johnson and Lindenstrauss [1984] that N points in high-dimensional space can be randomly projected, with high (asymptotic) probability, to a much smaller dimension without losing essential information.

Practically, various methods have been proposed to represent the sketching matrix, S [Drineas and Mahoney, 2016; Mahoney, 2011]. For example, the sketching matrix can be represented by independent identically distributed (i.i.d.) Gaussian random variables, or i.i.d. random variables drawn from any sub-Gaussian distribution [Drineas and Mahoney, 2016], or even represented by a product of two matrices, a random diagonal matrix with +1 or -1 on each diagonal entry, each with probability 1/2, and the Hadamard-Walsh matrix [Ailon and Chazelle, 2010]. In this work, we employ the randomization matrix scheme similar to the one used in Le et al. [2015] because of its stronger conditioning properties than other sketching matrix [Drineas and Mahoney, 2016]. In particular, the components of sketching matrix follow a Gaussian normal distribution (mean 0 and variance 1). The idea behind employing the randomization matrix as the sketching matrix

follows the one described in *Dasgupta and Gupta* [2003] and *Achlioptas* [2003]. It has been demonstrated in those references that such a selection of sketching matrix will be able to precondition arbitrary input matrices so that uniform sampling in the randomly rotated basis yields comparable performance to non-uniform sampling in the original basis.

3.3. Randomized Geostatistical Inversion Algorithm

To summarize our new randomized geostatistical inversion algorithm, we provide a detailed description of the algorithm below.

Input: k_{red} , ξ_0 , and β_0 , $IterCount_{\text{max}}$;

Output: $\mathbf{m}^{(k)}$

- 1: Initialize $found = \text{false}$;
- 2: Initialize k_{red} , ξ_0 , and β_0 ;
- 3: Generate the sketching matrix according to Sec. 3.2;
- 4: Obtain the data-reduced problem according to Eq. (16);
- 5: Update the data covariance matrix R' according to Eq. (17);
- 6: **while** $\{(\text{not } found) \text{ and } (IterCount < IterCount_{\text{max}})\}$ **do**
- 7: Solve for the solution of the reduced linear system in Eq. (15);
- 8: Update the iterate according to Eq. (18);
- 9: **end while**

Algorithm 1: Randomized Geostatistical Approach (RGA)

It would be worthy to mention that both direct linear solvers or iterative solvers can be utilized to solve the reduced linear systems in Eq. (15). Considering in most cases, the reduced linear systems usually yields relative small system matrices, we employ a direct solver to solve the reduced linear systems.

4. Computational and Memory Cost Analysis

To better understand the cost of our new randomized geostatistical inversion algorithm, we provide both the computational and memory cost analysis of our method described in Alg. 1. To set up the problem, we assume that the number of model parameters is \tilde{m} , the number of observations is \tilde{n} , hence the size of the Jacobian matrix $H \in \mathcal{R}^{\tilde{n} \times \tilde{m}}$ and the covariance matrix $Q \in \mathcal{R}^{\tilde{n} \times \tilde{n}}$. We also assume the rank of the sketching matrix is k_{red} . The polynomial matrix $\xi \in \mathcal{R}^{\tilde{m} \times \tilde{p}}$. As a reference method, we select the method of PCGA, which is developed in *Kitanidis and Lee* [2014] and *Lee and Kitanidis* [2014].

4.1. Computational Cost

Considering most of the numerical operations in Alg. 1 involve only matrix and vector operations, we use the floating point operations per second (FLOPS) and the big- \mathcal{O} notation to quantify the computational cost [Golub and Van Loan, 1996]. In numerical linear algebra, BLAS operations are categorized into three levels. Level-1 operations involve an amount of data and arithmetic that is linear in the dimension of the operation. Those operations involving a quadratic amount of data and a quadratic amount of work are Level-2 operations [Golub and Van Loan, 1996]. Following this notation, vector dot-product, addition and subtraction are examples of BLAS Level-1 operations (BLAS 1). Matrix-vector multiplication is a BLAS Level-2 operation (BLAS 2). Matrix-matrix multiplication is a BLAS Level-3 operation (BLAS 3).

Again assuming we employ QR factorization to solve the linear system in Eq. (12), the total computational cost will be

$$\text{COMP}_{\text{PCGA}} \approx \mathcal{O}(2 \cdot (\tilde{m} + \tilde{p})^3) + \mathcal{O}(2 \cdot (\tilde{m} + \tilde{p})^2) + \mathcal{O}((\tilde{m} + \tilde{p})^2), \quad (19)$$

where the first term corresponds to the cost of QR factorization, the second term is the cost to form the right hand side, and the last term is the cost to perform the back substitution.

On the other hand, the computational cost of our new RGA can be derived

$$\text{COMP}_{\text{RGA}} \approx \mathcal{O}(2 \cdot (k_{\text{red}} + \tilde{p})^3) + \mathcal{O}(2 \cdot (k_{\text{red}} + \tilde{p})^2) + \mathcal{O}((k_{\text{red}} + \tilde{p})^2). \quad (20)$$

Even though the dominating computational cost of our method and standard PCGA are both BLAS 3, our method can be significantly more efficient, because we can choose $k_{\text{red}} \ll \tilde{m}$ for problems with many observations. Comparing Eq. (20) to Eq. (19), the total cost of our method takes about $(\mathcal{O}(k_{\text{red}})/\mathcal{O}(\tilde{m}))^3$ to the cost of the PCGA method.

It should be noted here that this analysis explores the computational cost of the linear algebra associated with performing an iteration of the inverse analysis. Another significant computational cost comes from solving the forward model repeatedly. However, when PCGA is used and \tilde{m} is sufficiently large, the computational cost associated with these linear algebra operations dominates the cost of running the forward model repeatedly. By reducing the cost of this linear algebra, RGA results in a situation where the computational cost of repeatedly solving the forward model is the dominant cost in the inverse analysis.

4.2. Memory Cost

Both of our new algorithm in Alg. 1 and the reference method PCGA discussed in *Kitanidis and Lee* [2014] and *Lee and Kitanidis* [2014] rely on dense matrix storage. Hence, the major memory cost is used to store the matrices. Out of all these matrices, the largest matrix required to store is the system matrix in Eq. (12) for the PCGA method or the matrix in Eq. (15) for our method. Provided with the setup of the problem size, the

dimension of system matrix in Eq. (12) is $\mathcal{R}^{(\tilde{m}+\tilde{p}) \times (\tilde{m}+\tilde{p})}$. Assuming QR factorization is used as the direct solver to solve the linear system in Eq. (12), an orthogonal and an upper-triangular matrix will be further obtained. Therefore, the total memory cost will be

$$\text{MEM}_{\text{PCGA}} \approx \gamma \cdot (\tilde{m} + \tilde{p}) \times (\tilde{m} + \tilde{p}) \cdot \text{BYTES}, \quad (21)$$

where $\gamma \approx 3$ for the method of PCGA and BYTES are the number of bits depending on numeric precision of the computing hardware.

Similarly, we can also calculate the dimension of the corresponding linear system in Eq. (15) for our method. Provided with a rank k_{red} sketching matrix, the dimension of the resulting linear system is $\mathcal{R}^{(k_{\text{red}}+\tilde{p}) \times (k_{\text{red}}+\tilde{p})}$. Therefore, the memory cost will be

$$\text{MEM}_{\text{RGA}} \approx \gamma \cdot (k_{\text{red}} + \tilde{p}) \times (k_{\text{red}} + \tilde{p}) \cdot \text{BYTES}. \quad (22)$$

Comparing Eq. (22) to Eq. (21), we see that the memory cost of our method is only approximately $\kappa \approx (k_{\text{red}}/\tilde{m})^2$ of that of the PCGA method. Through our tests below, we show that for most situations, the ratio can be $\kappa \approx 1.0\%$ or even smaller.

5. Numerical Results

In this section, we provide numerical examples to demonstrate the efficiency of our new randomized geostatistical inversion algorithm. The reference problem is a transient groundwater equation. For the purposes of calibration, the hydraulic head were “observed” from a solution of the groundwater equation using a reference transmissivity field with the addition of noise.

To have a comprehensive comparison, we provide three sets of tests. In Sec. 5.1, we provide the convergence test of our method. In Sec. 5.2, we report the performance of our

method as a function of the number of rows, k_{red} , in the sketching matrix. In Sec. 5.3, we test our method on inverse problems with an increasing number of measurements up to 10^7 . We denote our method based on the randomization matrix as “RGA”. We denote the method developed in *Lee and Kitanidis* [2014] and *Kitanidis and Lee* [2014] as “PCGA”.

We select Julia as our programming tool because of its efficiency and simplicity. Julia is a high-level programming language designed for scientific computing [Bezanson *et al.*, 2014]. The Julia code for our RGA algorithm is available as a part of the open-source release of Julia version of MADS (Model Analysis and Decision Support) at “<http://mads.lanl.gov/>” [Vesselinov *et al.*, 2015]. For the methods of the QR factorization and fundamental BLAS operations are all implemented using the system routines provided in the Julia packages. As for the computing environment, we run the first two sets tests on a computer with 40 Intel Xeon E5-2650 cores running at 2.3 GHz, and 64 GB memory, and the last set of tests on a higher-memory machine with 64 AMD Opteron 6376 cores running at 2.3 GHz and 256 GB of memory.

The stopping criterion is an important issue for any iterative method including our method. In our work, we employ two stopping criteria shown below to justify the convergence of the iteration

$$\|\mathbf{m}^{(k+1)} - \mathbf{m}^{(k)}\|_2^2 / \|\mathbf{m}^{(k)}\|_2^2 \leq \text{TOL}, \quad (23)$$

and

$$k \leq k_{\text{MAX}}, \quad (24)$$

where $\text{TOL} = 10^{-6}$ and $k_{\text{MAX}} = 50$. If either Eq. (23) or Eq. (24) is satisfied, the iteration procedure will stop.

5.1. Test of the Convergence

In our first numerical example, we first test the convergence of our new methods. The reference model is solved on a grid containing 100×100 , pressure nodes and a total of 10100 model parameters (100×101 log-transmissivities along x axis, 101×100 log-transmissivities along y axis). We generate a ground truth, which is shown in Fig. 3(a). We utilize the variance ($\sigma_{\mathbf{m}}^2$) and an exponent ($\beta_{\mathbf{m}}$ – related to the fractal dimension of the field and the power-law of the field’s spectrum) to characterize the heterogeneity of the considered fields [Peitgen and Saupe, 1988]. In this example, we set the variance $\sigma_{\mathbf{m}}^2 = 0.5$ and power $\beta_{\mathbf{m}} = -3.5$. The number of the measurements generated in this test is 16,000. These measurements come from running the transient simulation to simulate 4 different pumping tests. In each test, 1000 hydraulic head observations are recorded at each of 4 different wells.

We illustrate one of the randomization matrices in Fig. 2. The rank of the randomization matrix is $k_{\text{red}} = 256$. The elements of the randomization matrix follow a normal distribution with mean 0 and standard deviation 1. Because of the width limitation of the page, we only show the first 1000 columns of the randomization matrix.

Fig. 3(b) illustrates the result using the PCGA method. Our method yields the results in Fig. 3(c). Comparing to the true model in Fig. 3(a), our method obtains a good result, representing both the high and low log-permeability regions. Visually, our method yields a comparable result to the one obtained using PCGA method in Fig. 3(b).

To further quantify the inversion error of different inverse modeling methods, we calculate both the relative-model-error (RME) and relative-data-error (RDE) of the inversion

results

$$\text{RME}(\mathbf{m}) = \frac{\|\mathbf{m} - \mathbf{m}_{\text{ref}}\|_2}{\|\mathbf{m}_{\text{ref}}\|_2}, \quad (25)$$

where \mathbf{m} is the inversion and \mathbf{m}_{ref} is the ground truth.

$$\text{RDE}(\mathbf{d}) = \frac{\|\mathbf{d} - \mathbf{d}_{\text{rec}}\|_2}{\|\mathbf{d}_{\text{rec}}\|_2}, \quad (26)$$

where \mathbf{d} is the simulated data based on the inversion and \mathbf{d}_{rec} are the recorded observations (which comes from solving the forward model for the reference transmissivity field and adding noise).

We provide the plot of the rates of convergence of the PCGA method and our RGA method in Fig. 4. We observe that both our method and the PCGA yield a very similar rate of the convergence as a function of the number of iterations steps. At each iteration, these methods yield similar relative data error and model error values. When both methods converged, the RME values of our RGA method and PCGA method are 56.3% and 64.5%, respectively. Therefore, together with the inversion result in Fig. 3, we demonstrate that our RGA method yields a comparable accuracy to the PCGA method in this situation where both methods can be applied. We note, however, that one of the main benefits of the RGA method is that it can applied in situations with a very large number of observations that result in the PCGA method running out of memory. Also it is worth mentioning that RGA is much more computationally efficient than the PCGA method. In particular, PCGA took about 32,000 seconds to converge, while it took only about 1,020 seconds for RGA to converge – a speed-up ratio of ~ 31 .

5.2. Test on the Rank of the Sketching Matrix

The rank of the randomization matrix k_{red} can be critical to the accuracy and efficiency of our RGA method. In this section, we test our algorithm using randomization matrix with different rank values. The values of k_{red} used in the problem are 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, and 8192.

In Fig. 5, we further provide both the RME value defined in Eq. (25) in Fig. 5(a) and RDE value defined in Eq. (26) in Fig. 5(b). We notice that the larger k_{red} becomes, the smaller the error of the inversion. In the first few selections of k_{red} , there is significant decrease of the RME values, which means that the inversion results are improving. In particular, the inversion results are completely off when the k_{red} is 4. At the selection of $k_{\text{red}} = 256$, the RME curve starts to level off while RDE curve still reduces.

Figure 6 shows the corresponding wall time cost of different values of k_{red} . It can be observed that the time is quite stable around 500 seconds until $k_{\text{red}} = 2048$, where the CPU time increases to about 550 seconds. When $k_{\text{red}} = 8192$, the CPU time cost is 2902 seconds. This can be explained by the fact that when k_{red} is relatively small, the CPU time is mostly dominated by the forward modeling operations, while as k_{red} increases, the linear solver for the solution of the system in Eq. (15) starts to dominate. Even though the data misfit of the inversion becomes smaller as the increase of the k_{red} , hardly any useful information is introduced into the results.

From this test, we conclude that the optimal selection of the k_{red} value is ranging from 256 to 1024 considering the factors including model error, data misfit, as well as the corresponding time cost. In general, when choosing the value of k_{red} , one would want to choose a value that is large enough to produce accurate results (i.e., large enough to be in the flat portion of Fig. 5(a)) and small enough so that the method is computationally

efficient (i.e., small enough to be in the flat portion of Fig. 6). The Johnson-Lindenstrauss Lemma provides some *a priori* guidance on the former while the latter can be guided by a back-of-the-envelope calculation relating the cost of the forward model solve to the cost of the linear algebra.

5.3. Test on the Number of Observations

To better understand the scalability of our method, we test RGA on a set of inverse problems that have an increasing number of observations. The number of observations have been increased to simulate the number of observations in the practical situations such as the one illustrated in Fig. 1. Specifically, we test our algorithm on inverse problems where the number of observations is equal to 2.56×10^5 , 6.25×10^5 , 1.296×10^6 , 2.401×10^6 , 4.096×10^6 , 6.561×10^6 , and 1.0×10^7 . As before, the observations come from simulating a series of pumping tests and recording “observations” at a number of monitoring wells. For each observation well, there are 1000 observations for each pumping test. The increasing number of observations come from increasing the number of pumping tests and the number of observation wells. For example, the case with 2.56×10^5 observations involves 16 pumping tests and 16 observation wells while the case with 1.0×10^7 involves 100 pumping tests and 100 observation wells. The reference transmissivity field is same as the one as in Fig. 3(a). The value of k_{red} is again set to 256.

The number of observations precludes the possibility of using the PCGA method in *Lee and Kitanidis* [2014] and *Kitanidis and Lee* [2014] (the computer runs out of memory). Hence we are not able to provide the corresponding results obtained using PCGA even for the smallest number of observations, 2.56×10^5 . However, using RGA, we are still able to perform the inverse analysis with ten million observations. We tested our RGA

method on all the problem sizes mentioned above and provide the corresponding results where the number of observations is 2.56×10^5 , 4.096×10^6 , and 1.0×10^7 in Fig. 7. We notice that our RGA method yields reasonable inversion results when the size of the data sets becomes massive. As a comparison, the PCGA method fails in all three cases of Figs. 7(b), 7(c), and 7(d) because of the insufficient memory.

We also provide the wall time costs of our methods with different numbers of observations in Fig. 8. Shown in Fig. 8 is the wall time to perform the model calibration with RGA and the wall time to perform a single model run. These times are shown for problems where the number of observations is 2.56×10^5 , 6.25×10^5 , 1.296×10^6 , 2.401×10^6 , 4.096×10^6 , 6.561×10^6 , and 1.0×10^7 . For all these problems, which vary over two orders of magnitude, the time to perform the full model calibration takes 28 times as long as performing a single model run and this could be reduced further with more CPU cores. Also we notice that the computational cost of RGA scales well with the number of observations. Through this test, we conclude that our method yields a much higher computability than the PCGA method in calibrating models with a large number of observations.

6. Conclusion

We have developed a computationally efficient, scalable, and implementation-friendly randomized geostatistical inversion method, which is especially suitable for inverse modeling with a large number of observations. Our method, which we call the Randomized Geostatistical Approach (RGA), is built upon the Principal Component Geostatistical Approach (PCGA) developed by Kitandis and others. To overcome the issues of excessive memory and computational cost that arises when using PCGA with a large number of

observations, we incorporated a randomized matrix sketching technique into PCGA. The randomization method can be seen as a data-reduction technique, because it generates a surrogate system that has much lower dimension than the original problem.

Through our computational cost analysis, we show that this matrix sketching technique reduces both the memory and computational costs significantly. Comparing with the PCGA method, our RGA method yields a much smaller problem to solve when computing the next step in the iterative optimization process, therefore reducing both the memory and computational costs. We demonstrate through our numerical example that a speed-up ratio of 31 can be achieved by using our RGA method compared to the PCGA method. It is reasonable to conclude that the speed-up ratio can be much significant when the size of the data sets increases. As demonstrated in the paper, eventually PCGA method will fail because of the insufficient memory while our RGA method can be much more robust and yield reasonable results with massive number of data sets.

In summary, with an ever-increasing amount of data being assimilated into hydrologic models, there is a need to develop a hydrologic inverse method that is able to handle a large number of observations. Our RGA method addresses such a need. The contribution of our work is to incorporate a randomized numerical linear algebra technique into the PCGA method. Through both the computational cost analysis and the numerical tests, we show theoretically and numerically that our RGA method is computationally efficient and capable of solving inverse problems with $\mathcal{O}(10^7)$ observations using modest computational resources (approximately ten US dollars in the cloud). Therefore, it shows great potential for characterizing subsurface heterogeneity for problems with a large number of observations.

Our new algorithm RGA is coded in Julia and implemented in the MADS open-source high-performance computational framework (<http://mads.lanl.gov>). However, the implementation of RGA is relatively simple, and can be easily added to any existing code employing the PCGA algorithm.

Acknowledgments.

Youzuo Lin, Daniel O'Malley, Ellen B. Le, and Velimir V. Vesselinov were support by Los Alamos National Laboratory Environmental Programs Projects. In addition, Daniel O'Malley was supported by a Los Alamos National Laboratory (LANL) Director's Post-doctoral Fellowship, and Velimir V. Vesselinov was supported by the DiaMonD project (An Integrated Multifaceted Approach to Mathematics at the Interfaces of Data, Models, and Decisions, U.S. Department of Energy Office of Science, Grant #11145687). All the data are available from the authors upon request (ylin@lanl.gov).

References

- Achlioptas, D. (2003), Database-friendly random projections: Johnson-Lindenstrauss with binary coins, *Journal of computer and System Sciences*, 66(4), 671–687.
- Ailon, N., and B. Chazelle (2010), Faster dimension reduction, *Communications of the ACM*, 53, 97–104.
- Ambikasaran, S., J. Y. Li, P. K. Kitanidis, and E. Darve (2013), Large-scale stochastic linear inversion using hierarchical matrices, *Computational Geosciences*, 17(6), 913–927.
- Bebendorf, M. (2008), *Hierarchical Matrices: A Means to Efficiently Solve Elliptic Boundary Value Problems*, vol. 63, Springer, New York.

- 508 Bezanson, J., A. Edelman, S. Karpinski, and V. B. Shah (2014), Julia: A fresh approach
509 to numerical computing., <http://http://julialang.org/>.
- 510 Borm, S., L. Grasedyck, and W. Hackbusch (2003), Introduction to hierarchical matrices
511 with applications, *Eng. Anal. Boundary Elem.*, 5(27), 405–422.
- 512 Carrera, J., and S. P. Neuman (1986), Estimation of aquifer parameters under transient
513 and steady state conditions: 1. maximum likelihood method incorporating prior infor-
514 mation, *Water Resources Research*, (22), 199–210.
- 515 Carrera, J., A. Alcolea, A. Medina, J. Hidalgo, and L. J. Slooten (2005), Inverse problem
516 in hydrogeology, *Hydrogeology journal*, 13(1), 206–222.
- 517 Clarkson, K. L., and D. P. Woodruff (2009), Numerical linear algebra in the streaming
518 model, in *Proceedings of the forty-first annual ACM symposium on Theory of computing*,
519 pp. 205–214, ACM.
- 520 Constantine, P. G., E. Dow, and Q. Wang (2014), Active subspace methods in theory and
521 practice: Applications to kriging surfaces, *SIAM Journal on Scientific Computation*,
522 36(4), A1500–A1524.
- 523 Dasgupta, A., R. Kumar, and T. Sarlós (2010), A sparse Johnson-Lindenstrauss trans-
524 form, in *Proceedings of the forty-second ACM symposium on Theory of computing*, pp.
525 341–350, ACM.
- 526 Dasgupta, S., and A. Gupta (2003), An elementary proof of a theorem of Johnson and
527 Lindenstrauss, *Random structures and algorithms*, 22(1), 60–65.
- 528 Drineas, P., and M. W. Mahoney (2016), RandNLA: Randomized numerical linear algebra,
529 *Communications of the ACM*, 6(6), 80–90.

- Drineas, P., M. M. W., M. S., and T. Sarlos (2011), Faster least squares approximation,
Numerische Mathematik, 117, 219–249.
- Engl, H. W., M. Hanke, and A. Neubauer (1996), *Regularization of Inverse Problems*,
Kluwer Academic Publishers.
- Golub, G. H., and C. F. Van Loan (1996), *Matrix Computations*, The Johns Hopkins
University Press, third edition.
- Hansen, P. C. (1998), *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects
of Linear Inversion*, SIAM.
- Huang, L., J. Shin, T. Chen, Y. Lin, K. Gao, M. Intrator, and K. Hanson (2016), Breast
ultrasound tomography with two parallel transducer arrays, in *Proc. SPIE 9783, Medical
Imaging 2016: Ultrasonic Imaging, Tomography, and Therapy*, pp. 97,830C–97,830C–
12.
- Hunt, R. J., J. Doherty, and M. J. Tonkin (2007), Are models too simple? Arguments for
increased parameterization, *groundwater*, 45, 254–262.
- Illman, W. A., S. J. Berg, and Z. Zhao (2015), Should hydraulic tomography data be
interpreted using geostatistical inverse modeling? a laboratory sandbox investigation,
Water Resources Research, 51(5), 3219–3237, doi:10.1002/2014WR016552.
- Johnson, W. B., and J. Lindenstrauss (1984), Extensions of Lipschitz mappings into a
Hilbert space, *Contemporary mathematics*, 26(189-206), 1.
- Kane, D. M., and J. Nelson (2014), Sparser Johnson-Lindenstrauss transforms, *Journal
of the ACM (JACM)*, 61(1), 4.
- Kitanidis, P. (1997a), *Introduction to Geostatistics: Applications to Hydrogeology*,
Stanford-Cambridge program, Cambridge University Press.

Kitanidis, P. K. (1995), Quasi-linear geostatistical theory for inversing, *Water resources research*, *31*(10), 2411–2419.

Kitanidis, P. K. (1997b), The minimum structure solution to the inverse problem, *Water resources research*, *33*(10), 2263–2272.

Kitanidis, P. K., and J. Lee (2014), Principal component geostatistical approach for large-dimensional inverse problem, *Water Resources Research*, *50*, 5428–5443.

Krebs, J. R., J. E. Anderson, D. Hinkley, R. Neelamani, S. Lee, A. Baumstein, , and M. D. Lacasse (2009), Fast full-wavefield seismic inversion using encoded sources, *Geophysics*, *74*, WCC177–WCC188.

Le, E. B., A. Myers, and T. Bui-Thanh (2015), A randomized misfit approach for data reduction in large-scale inverse problems, *Submitted*.

Lee, J., and P. K. Kitanidis (2014), Large-scale hydraulic tomography and joint inversion of head and tracer data using the principal component geostatistical approach (pcga), *Water Resources Research*, *50*, 5410–5427.

Lin, Y., D. OMalley, and V. V. Vesselinov (2016), A computationally efficient parallel Levenberg-Marquardt algorithm for highly parameterized inverse model analyses, *Water Resources Research*, *52*, 6948–6977, doi:10.1002/2016WR019028.

Liu, X., Q. Zhou, J. T. Birkholzer, and W. A. Illman (2013), Geostatistical reduced-order models in underdetermined inverse problems, *Water Resources Research*, *59*, 6587–6600.

Liu, X., Q. Zhou, P. K. Kitanidis, and J. T. Birkholzer (2014), Fast iterative implementation of large-scale nonlinear geostatistical inverse modeling, *Water Resources Research*, *50*, 198–207.

- 575 Mahoney, M. W. (2011), Randomized algorithms for matrices and data, *Foundations and*
576 *Trends® in Machine Learning*, 3(2), 123–224.
- 577 Meng, M. A., X. Saunders, and M. W. Mahoney (2014), LSRN: A parallel iterative solver
578 for strongly over- or underdetermined systems, *SIAM J. Sci. Comput.*, 36, 95–118.
- 579 Moghaddam, P. P., H. Keers, F. J. Herrmann, and W. A. Mulder (2013), A new op-
580 timization approach for source-encoding full-waveform inversion, *Geophysics*, 78(3),
581 R125–R132.
- 582 Neuman, S. P., and S. Yakowitz (1979), A statistical approach to the inverse problem of
583 aquifer hydrology: 1. theory, *Water Resources Research*, 15(4), 845–860, doi:10.1029/
584 WR015i004p00845.
- 585 Neuman, S. P., G. E. Fogg, and E. A. Jacobson (1980), A statistical approach to the
586 inverse problem of aquifer hydrology: 2. case study, *Water Resources Research*, 16(1),
587 33–58.
- 588 Nowak, W., and O. A. Cirpka (2004), A modified Levenberg-Marquardt algorithm for
589 quasi-linear geostatistical inversing, *Advances in Water Resources*, 27, 737–750.
- 590 Paige, C. C., and M. A. Saunders (1975), Solution of sparse indefinite systems of linear
591 equations, *SIAM Journal of Numerical Analysis*, (12), 617–629.
- 592 Peitgen, H. O., and D. Saupe (1988), *The Science of Fractal Images*, Springer-Verlag New
593 York.
- 594 Rokhlin, V., and M. Tygert (2008), A fast randomized algorithm for overdetermined linear
595 least-squares regression, *Proc. Natl. Acad. Sci. USA*, 105(36), 13,212–13,217.
- 596 Saad, Y., and M. H. Schultz (1986), GMRES: A generalized minimal residual method for
597 solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.*, 3(7), 856–869.

- Saibaba, A. K., and P. K. Kitanidis (2012), Efficient methods for large-scale linear inversion using a geostatistical approach, *Water Resources Research*, 48(5), W05,522.
- Sarlos, T. (2006), Improved approximation algorithms for large matrices via random projections, in *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pp. 143–152, IEEE.
- Sun, N. (1994), *Inverse problems in groundwater modeling*, Kluwer Academic Publishers.
- Tarantola, A. (2005), *Inverse Problem Theory*, SIAM.
- Tonkin, M. J., and J. Doherty (2005), A hybrid regularized inversion methodology for highly parameterized environmental models, *Water Resources Research*, 41, W10,412.
- Vesselinov, V., S. Neuman, and W. Illman (2001a), Three-dimensional numerical inversion of pneumatic cross-hole tests in unsaturated fractured tuff 1. Methodology and borehole effects, *Water Resources Research*, 37(12), doi:10.1029/2000WR000133.
- Vesselinov, V., S. Neuman, and W. Illman (2001b), Three-dimensional numerical inversion of pneumatic cross-hole tests in unsaturated fractured tuff 2. Equivalent parameters, high-resolution stochastic imaging and scale effects, *Water Resources Research*, 37(12), doi:10.1029/2000WR000135.
- Vesselinov, V. V., D. O'Malley, Y. Lin, et al. (2015), MADS.jl: (Model Analyses and Decision Support) in Julia, <http://mads.lanl.gov/>.
- Vogel, C. (2002), *Computational Methods for Inverse Problems*, SIAM.
- Wang, K., T. Matthews, F. Anis, C. Li, N. Duric, and M. A. Anastasio (2015), Waveform inversion with source encoding for breast sound speed reconstruction in ultrasound computed tomography, *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 62(3), 475–493, doi:10.1109/TUFFC.2014.006788.

- 621 Woodruff, D. P. (2014), Sketching as a tool for numerical linear algebra, *arXiv preprint*
622 *arXiv:1411.4357*.
- 623 Yeh, T.-C. J., and J. Simunek (2002), Stochastic fusion of information for characterizing
624 and monitoring the vadose zone, *Vadose Zone*, (1), 2095–2105.
- 625 Zhang, J., and T.-C. J. Yeh (1997), An iterative geostatistical inverse method for steady
626 flow in the vadose zone, *Water Resources Research*, 33(1), 63–71.
- 627 Zhang, Z., L. Huang, and Y. Lin (2012), Efficient implementation of ultrasound wave-
628 form tomography using source encoding, in *Proc. SPIE 8320, Medical Imaging 2012:*
629 *Ultrasonic Imaging, Tomography, and Therapy*, pp. 832,003–832,003–10.

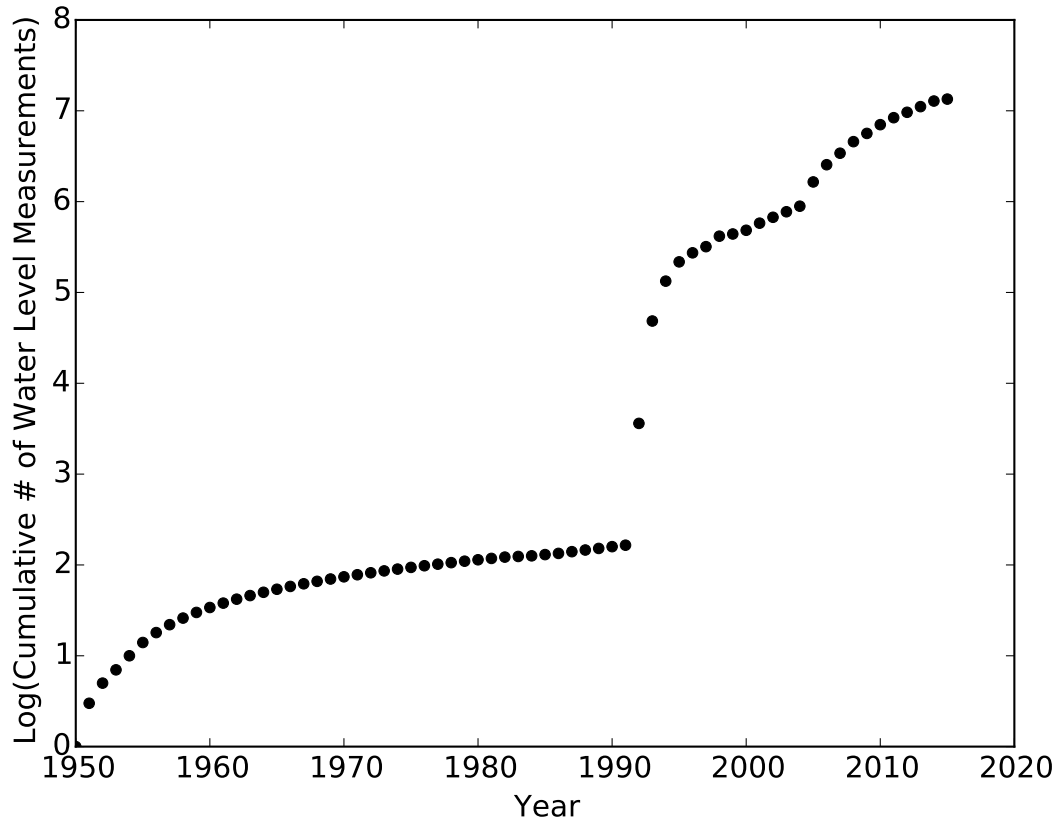


Figure 1. Cumulative number of water level measurements as a function of time collected at the Los Alamos National Laboratory site. These data provide important information about hydrogeologic site conditions and are included in various model analyses. The data are characterized by periodic, rapid increases in the rate of data collection which has produced a large data set with $\sim 10^7$ observations.

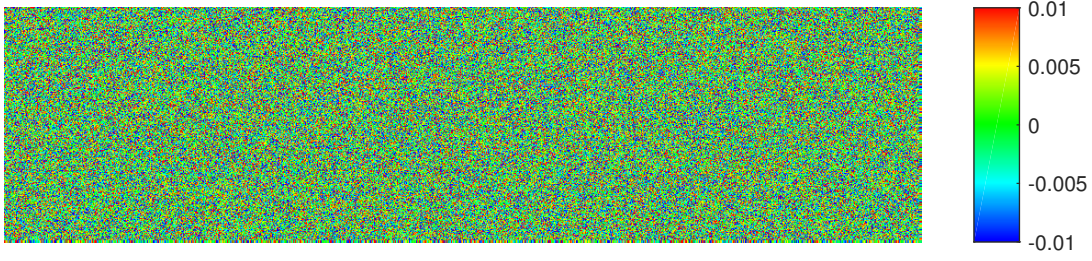


Figure 2. Illustration of an example randomization matrix used in the presented analyses with rank $k_{\text{red}} = 256$. The elements of the randomization matrix follow a normal distribution with mean 0 and standard deviation 1. Because of the width limitation of the page, we only show the first 1000 columns of the randomization matrix.

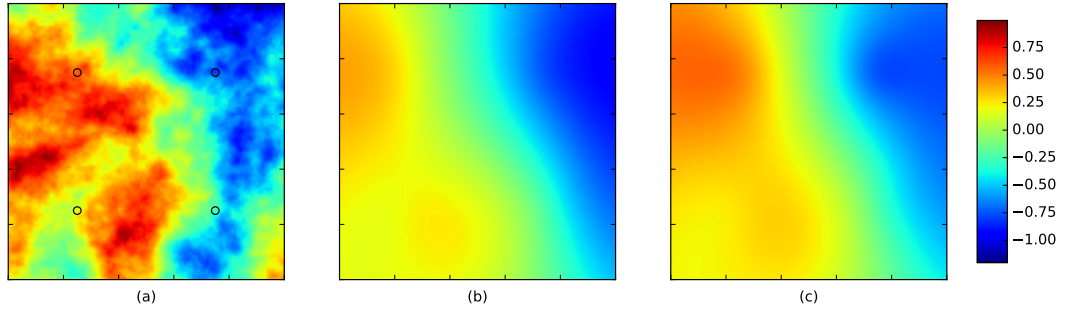
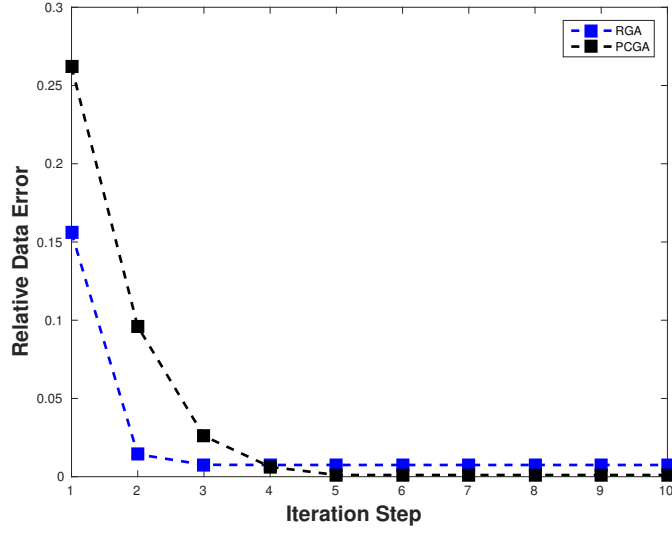
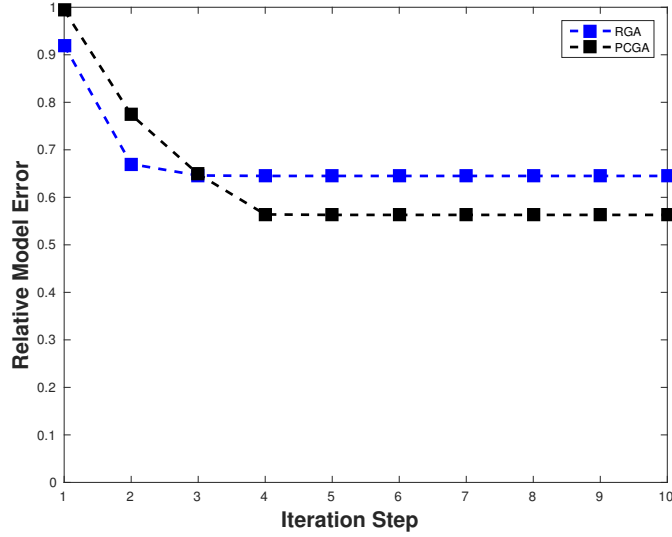


Figure 3. Synthetic log-transmissivity field (a) with variance $\sigma_{\mathbf{m}}^2 = 0.5$ and power $\beta_{\mathbf{m}} = -3.5$. Hydraulic conductivity and hydraulic head observation locations are indicated with circles. The results of the inverse modeling solved by PCGA (b) and our RGA algorithm (c) are shown. They are visually similar to each other. The RME values of the results in (b) and (c) are 56.3% and 64.5%, respectively. Hence, our RGA method yields comparable result to that obtained using the PCGA method.

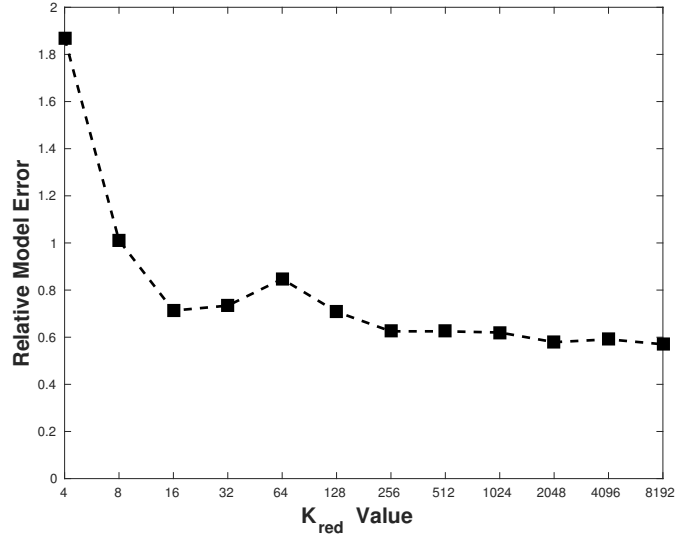


(a)

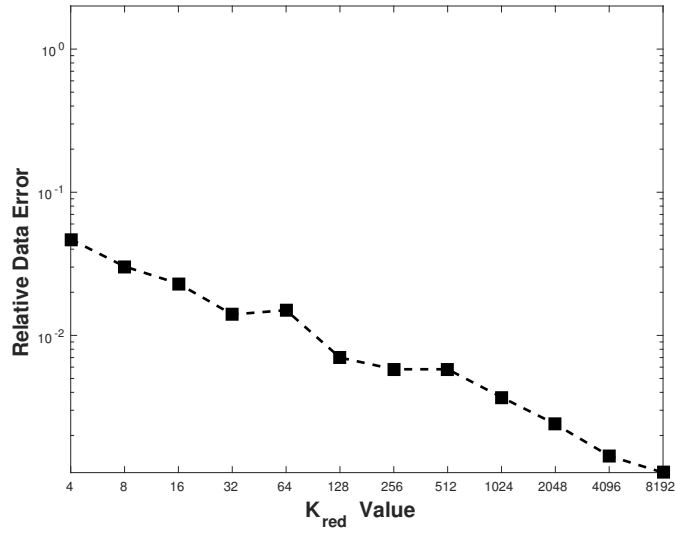


(b)

Figure 4. Convergence of the PCGA (in black) and our RGA (in blue) algorithms in terms of iteration steps. The rates of convergence for these two methods are very close to each other. However, the computational time of two methods to reach convergence are very different. In this case, PCGA converged for about 32,000 seconds, while RGA convergence took only 1,020 seconds. The RGA speed-up is about 31 times.



(a)



(b)

Figure 5. RME (a) and RDE (b) curves as defined in Eq. (25) and (26), respectively.

For k_{red} increasing from 4 to 256, there is a significant decrease in RME. For $k_{\text{red}} \geq 256$, the RME curve starts to level off while RDE curve still reduces.

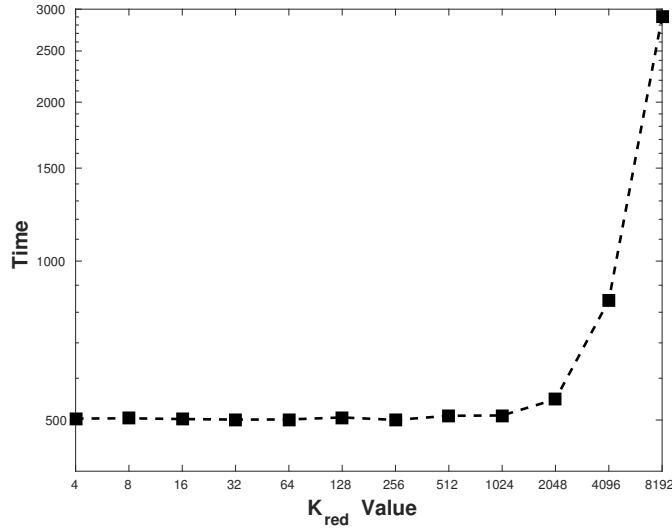


Figure 6. CPU time cost as a function of k_{red} . The CPU time is quite stable around 500 seconds for $k_{\text{red}} \leq 1024$. The time cost dependency on k_{red} can be explained by the fact that when k_{red} is relatively small, the CPU time is mostly dominated by the forward modeling operation, while as k_{red} increases, the linear solver for the solution of the system in Eq. (15) starts to dominate.

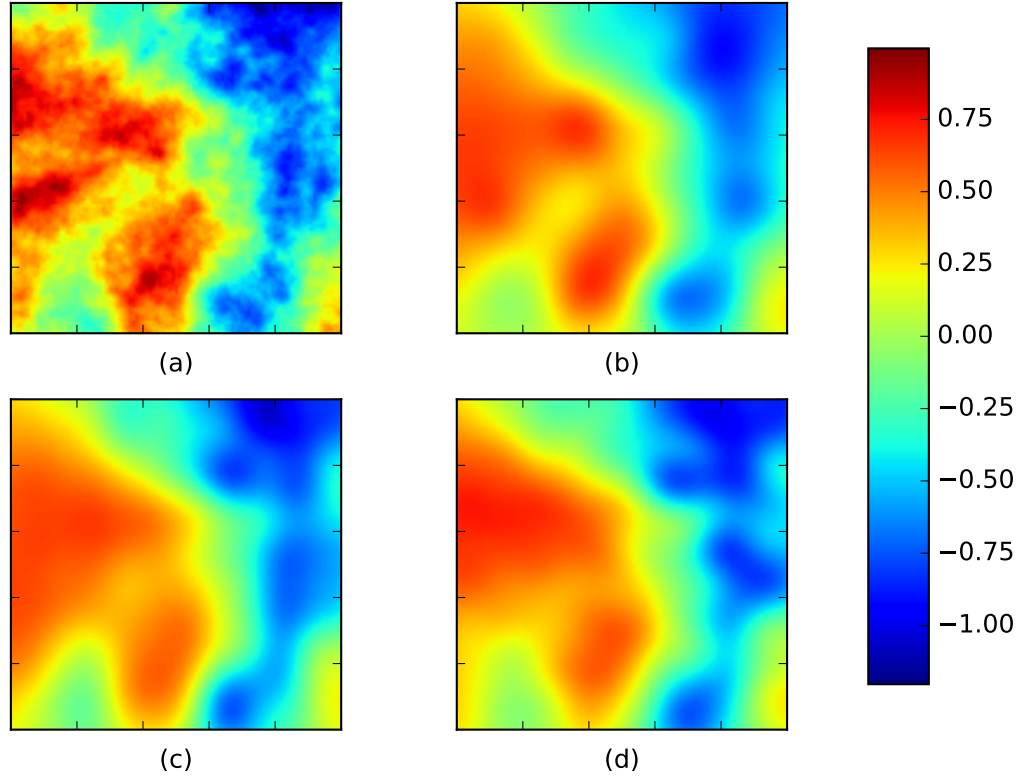


Figure 7. The “true” field (a) and inversion results of our RGA method with different numbers of observations including 2.56×10^5 (b), 4.096×10^6 (c) and 1.0×10^7 (d). Our RGA method yields reasonable inversion results when the size of the data sets becomes massive. As a comparison, the PCGA method fails in all three cases of (b), (c), and (d) because of the insufficient memory.

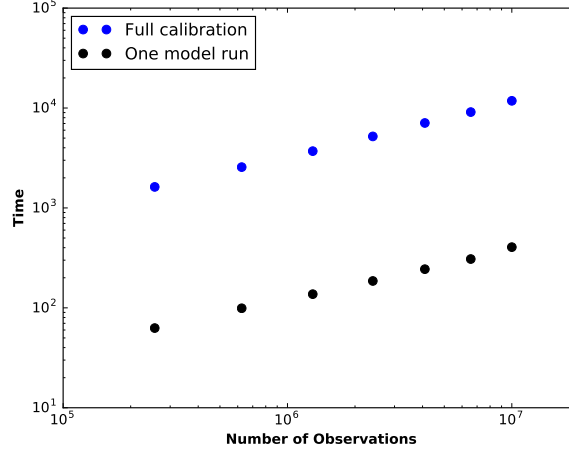


Figure 8. Wall-clock times to perform the model calibration with our RGA method and to perform a single model run. These times are shown for inverse analyses where the number of observations is 2.56×10^5 , 6.25×10^5 , 1.296×10^6 , 2.401×10^6 , 4.096×10^6 , 6.561×10^6 , and 1.0×10^7 . For all these analyses, which vary over two orders of magnitude, the time to perform the full model calibration takes 28 times as long as performing a single model run and this could be reduced further with more CPU cores.